

# Voting

## Introducere

Acest proiect reprezinta un sistem electronic de vot creat pentru a simplifica si securiza procesul de autentificare si exprimare a votului. Sistemul permite verificarea utilizatorilor pe baza unor carduri individuale si ofera posibilitatea de a vota rapid intre doua optiuni disponibile, centralizand automat rezultatele finale.

Scopul proiectului este de a demonstra modul in care un proces de vot poate fi realizat intr-un mod eficient, organizat si usor de utilizat, reducand posibilitatea erorilor si a voturilor multiple. Ideea de la care am pornit a fost necesitatea unui sistem accesibil care sa poata valida identitatea alegatorilor si sa ofere un raspuns imediat privind dreptul de acces la vot.

## Descriere generală



- Sistemul implementat reprezinta o platforma electronica de vot care permite autentificarea utilizatorilor si exprimarea votului intr-un mod rapid si securizat. Fiecare utilizator utilizeaza un card RFID/NFC pentru identificare, iar sistemul verifica automat daca acesta exista in lista de votanti eligibili incarcata anterior de pe calculator.
- Dupa validare, utilizatorul primeste un raspuns vizual si sonor care indica daca accesul la vot este permis sau respins. In cazul autentificarii cu succes, alegatorul poate selecta una dintre cele doua optiuni disponibile folosind butoanele fizice.
- Sistemul gestioneaza automat procesul de validare, prevenirea votului multiplu, inregistrarea voturilor si afisarea rezultatelor finale pe display. Interactiunea dintre componentele hardware si software este coordonata de microcontroller, care controleaza citirea datelor, logica de votare si semnalele de feedback.

## Hardware Design

Lista de piese:

- Placa de dezvoltare ATmega328P Xplained Mini: Microcontrollerul principal care gestioneaza logica votarii, validarea ID-urilor si controlul perifericelor.
- Modul LCD 1602 cu Backlight Albastru de 3.3 V cu interfata I2C: Ecran utilizat pentru afisarea intrebarilor, instructiunilor de vot si a rezultatelor finale.

- Modul RFID RC522 13.56MHz cu card si tag prin SPI: Cititor utilizat pentru scanarea cardurilor de identitate si autentificarea nominala a alegatorilor.
- Carduri sau Tag-uri RFID/NFC: Suporturi de date individuale folosite de alegatori pentru a se identifica in mod unic in sistem.
- 2 Butoane Fizice: Componente pentru input utilizate de alegatori pentru a-si exprima optiunea de vot prin apasare.
- Modul buzzer pasiv KY-006: Utilizat pentru confirmarea sonora a scanarii cardului sau semnalizarea incercarilor de vot multiplu.
- LED-uri (Verde si Rosu): Indicatori vizuali pentru starea procesului (acces permis, respectiv acces respins sau eroare).
- Rezistente de Pull-down/Pull-up: Componente pasive folosite pentru a asigura citirea corecta a starilor butoanelor fizice.
- Breadboard 400 puncte: Suport pentru realizarea conexiunilor hardware si prototiparea rapida a circuitului fara lipire.
- Fire Dupont Tata-Tata si Tata-Mama: Conductorii utilizati pentru legaturile electrice intre placa, senzori, butoane si afisaj.
- Cablu de date (USB): Cablu utilizat pentru programarea microcontrolerului si comunicarea UART cu PC-ul pentru configurarea listei de votanti.

Placa este alimentată direct prin cablul USB. Pinul GND al plăcuței este conectat la șina de GND a breadboard-ului, iar pinul de 5V este conectat la șina de VCC a breadboard-ului.

Butoanele sunt configurate hardware în mod pull-down, utilizând rezistoare externe de 10 kΩ conectate între pinul de semnal și masă (GND). Astfel, în stare de repaus, microcontroller-ul citește un nivel logic LOW stabil, iar la apăsarea butonului circuitul se închide, generând un semnal HIGH pe pinii PC0 și PC1.

LED-urile sunt configurate astfel încât un nivel logic HIGH le aprinde, iar LOW le stinge. Între VCC și LED-uri este conectată o rezistență de limitare a curentului, utilizată pentru protecție.

Modulul RFID RC522 este conectat direct la placă, fără a utiliza breadboard-ul. Acesta funcționează la 3.3V, nu la 5V.

Pentru LCD, deoarece nu a fost utilizat un potențiomtru, am facut un divizor de tensiune (foloseam pwm dar acesta dadea un puls instabil si uneori apare gibberish pe lcd) Modul LCD 1602

<b>Pin componenta</b>	<b>Legatura</b>
VSS	GND breadboard
VDD	VCC breadboard
V0	div tensiunde format din 1 rez 1k 2 rez 380 si o rez 10k
RS	PD7 microcontroller
RW	GND breadboard
E	PB0 microcontroller
D4	PD2 microcontroller

D5	PD3 microcontroller
D6	PD4 microcontroller
D7	PD5 microcontroller
A	23b breadboard
K	GND breadboard
rez 380 ohm	23a - VCC breadboard

## RFID-RC522

Pin componenta	Legatura
SDA	PB2 microcontroller
SCK	PB5 microcontroller
MOSI	PB3 microcontroller
MISO	PB4 microcontroller
GND	GND microcontroller
RST	PB1 microcontroller
3V3	3V3 microcontroller

## buzzer pasiv KY-006

Pin componenta	Legatura
minus (-)	GND breadboard
VCC	VCC breadboard
S	PC5 microcontroller

## BTN1

Pin componenta	Legatura
leg 1	PC0 microcontroller
leg 2	GND breadboard
rezistenta 10k ohm	leg1 - VCC breadboard

## BTN2

Pin componenta	Legatura
leg 1	PC1 microcontroller
leg 2	GND breadboard
rezistenta 10k ohm	leg1 - VCC breadboard

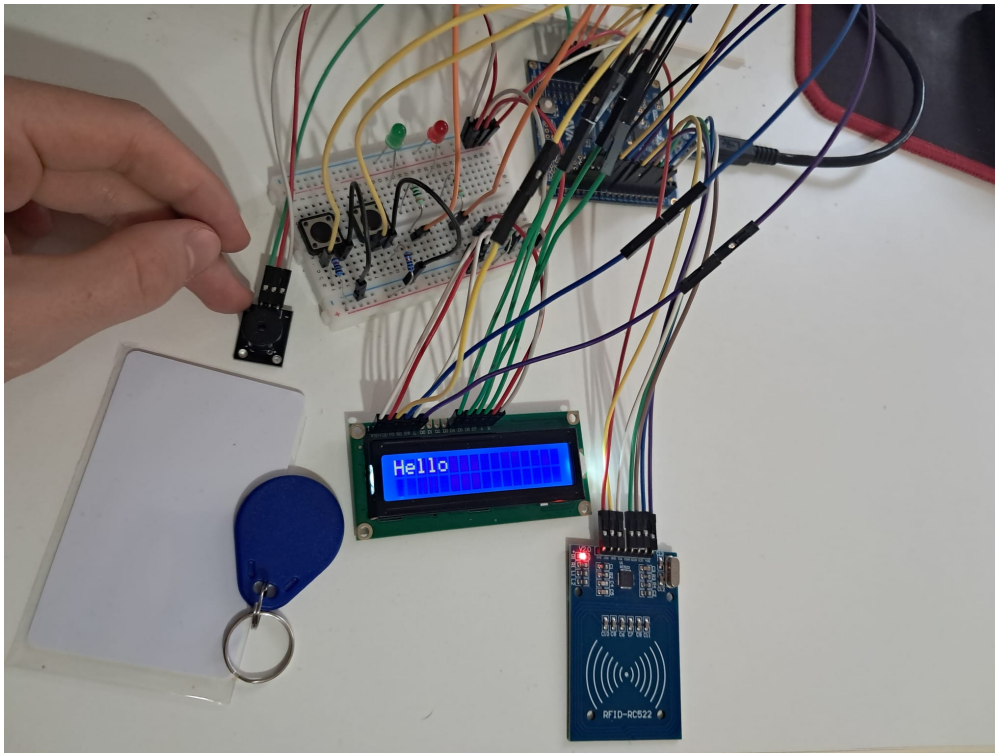
## LED VERDE

Pin componenta	Legatura
minus (-)	GND breadboard
plus (+)	j14
rez 380 ohm	f14 - PC3 breadboard

## LED ROSU

Pin componenta	Legatura
----------------	----------

minus (-)	GND breadboard
plus (+)	j20
rez 380 ohm	f20 - PC2 breadboard



## Software Design

- Demo: <https://youtube.com/shorts/VWpsEvxI2Nw>
- github: <https://github.com/dabijaAlex/Voting>

Proiectul a fost dezvoltat in PlatformIO, folosind framework-ul Arduino pentru microcontrolerul ATmega328P. Codul este scris in C++ si este complet modularizat pentru a asigura o structura curata si o mentenanta usoara.

Sistemul utilizeaza intrari si iesiri digitale (GPIO) pentru controlul LED-urilor de stare, butoanelor de vot (YES/NO) si al buzzerului. De asemenea, firmware-ul integreaza comunicatia hardware pe magistrala SPI pentru cititorul de carduri RFID, bit-banging software pe 4 biti pentru ecranul LCD paralel si comunicatia UART (USART0) prin Serial Monitor pentru configurare, control si logarea cardurilor neautorizate.

Arhitectura modulara a codului:

- **config.h**: Centralizeaza toate definitiile de pini hardware (LED-uri, buzzer, butoane, pini SPI si LCD) si constantele globale (debounce, dimensiuni maxime).
- **global\_vars.h / global\_vars.cpp**: Gestioneaza alocarea de memorie si partajarea sigura a variabilelor de stare, vectorilor de votanti si obiectelor prin directiva extern.
- **lcd.h / lcd.cpp**: Contine driverul grafic de control pentru ecranul paralel si algoritmul inteligent de

wrapping al textului.

- **audio.h / audio.cpp**: Gestioneaza semnalele acustice emise de buzzer pentru interactiunea cu utilizatorul.
- **utils.h / utils.cpp**: Include logica de parsare a string-urilor Hex, verificarea semnaturilor UID si initializarea initiala a perifericelor.
- **main.cpp**: Reprezinta orchestratorul principal ce contine rutinele de intrerupere si masina de stari.

Functii principale implementate:

Funcție	Rol
main()	Punctul de intrare obligatoriu in aplicatia C++. Initializeaza timerele si registrii interni prin functia Arduino <code>init()</code> si ruleaza intr-o bucla infinita programul.
setup()	Configureaza starea initiala a perifericelor hardware, porneste comunicatia seriala, magistrala SPI, cititorul RFID si ecranul LCD, aducand sistemul intr-o stare stabila de functionare.
loop()	Reprezinta bucla principala a programului. Verifica asincron buffer-ul serial si ruleaza logica aplicatiei prin intermediul unui automat cu 9 stari logice distincte.
ISR(PCINT1_vect)	Rutina de intrerupere pe schimbarile de pin (Port C). Gestioneaza debouncing-ul software prin timestamp-uri <code>millis()</code> pentru a inregistra corect apasarile butoanelor de vot YES si NO.
lcd_init()	Initializeaza fizic ecranul LCD paralel in modul de lucru pe 4 biti, trimitand secventele de initializare si setand parametrii de afisare (cursor dezactivat, directia de scriere).
lcd_command(uint8_t cmd)	Trimite o comanda de control catre registrul de instructiuni al LCD-ului (ex: curatare ecran, deplasare cursor) utilizand o abordare sigura cu blocarea temporara a intreruperilor.
lcd_data(uint8_t data)	Transmite un octet de date catre registrul RAM (DDRAM) al LCD-ului pentru a afisa un caracter specific pe ecran la pozitia curenta a cursorului.
lcd_display_message()	Afiseaza mesaje pe ecranul LCD. Include un algoritm inteligent de wrapping care cauta ultimul spatiu ( ' ' ) din primele 16 caractere pentru a muta textul pe randul doi fara a rupe cuvintele.
sound_buzzer()	Genereaza o frecventa bruta pe pinul buzzerului, comutand starea logica HIGH/LOW la un interval specific de microsecunde pentru o durata determinata de milisekunde.
play_accept_sound()	Ruleaza o secventa acustica scurta si ascutita pentru a confirma vizual si auditiv ca un card valid a fost scanat si accesul la vot a fost permis.
play_denied_sound()	Genereaza un ton grav si prelungit pentru a semnala o eroare critica sau incercarea de utilizare a unui card neautorizat.
play_warning_sound()	Emite un semnal acustic intermitent de avertizare pentru a notifica utilizatorul ca un votant autorizat incearca sa introduca un buletin de vot duplicat.
check_uid_match()	Compara octet cu octet o semnatura UID scanata de la un card RFID cu baza de date locala a sesiunii pentru a determina validitatea acesteia.
clear_rfid_cache()	Curata complet structura interna de date a cititorului MFRC522 si buffer-ul UID pentru a pregati senzorul pentru o noua citire corecta.
parse_uid_string()	Convertecte string-ul ASCII de 8 caractere hexazecimale primit din Serial Monitor in octeti fizici de date pentru a putea fi stocati si comparati in memorie.

print_unauthorized_uid()	Formateaza si transmite prin USART0 catre Serial Monitor un mesaj de alerta ce contine ID-ul hexazecimal complet al unui card respins, permitand monitorizarea tentativelor de frauda.
--------------------------	--

Logica Automatului de Stari (State Machine): Sistemul tranzitioneaza dinamic intre 9 stări: STATE\_SLEEPING (asteapta comanda “start”), STATE\_SETUP\_QUESTION (salveaza intrebarea), STATE\_SETUP\_VOTERS (configureaza numarul si UID-urile votantilor), STATE\_SHOW\_WELCOME (solicita card), STATE\_WAIT\_FOR\_CARD (interogheaza RFID si blocheaza butoanele), STATE\_VOTING\_PROMPT (asteapta alegerea YES/NO), STATE\_ACCESS\_DENIED (card necunoscut), STATE\_ALREADY\_VOTED (vot duplicat) si STATE\_FINISHED (afisare rezultate finale).

Interceptia Globala de Reset: La fiecare iteratie din loop( ), sistemul verifica registrul hardware UCSR0A. Daca se detecteaza cuvantul cheie “reset” primit prin terminal, logica normala a starii curente este suprascrisa instantaneu. Sistemul curata toate datele din buffer, reseteaza flag-urile (inclusiv variabila globala de printare seriala printed\_final) si aduce sistemul inapoi in starea initiala de veghe (STATE\_SLEEPING).

Biblioteci folosite:

Biblioteca	Rol
Arduino.h	Functii de baza pentru interactiunea cu hardware-ul ATmega328P: control GPIO, gestionare timere, delay, macro-uri de bit.
SPI.h	Gestioneaza comunicatia pe magistrala hardware SPI (pinii MOSI, MISO, SCK) pentru transferul de date de mare viteza cu perifericele active.
MFRC522.h	Asigura driverul software complet pentru controlul cipului RFID MFRC522, acoperind initializarea antenei, citirea si incarcarea UID-ului.
usart.h	Modul extern dezvoltat in C dedicat interactiunii directe cu registrii USART0 (UBRR0, UCSR0B, UDR0) pentru transmisia si receptia pe interfata seriala.

## Resurse Hardware

- **Placa de dezvoltare ATmega328P Xplained Mini:** <https://www.microchip.com/en-us/development-tool/atmega328p-xmini>
- **Documentatie Laborator PM (Xplained Mini):** <https://ocw.cs.pub.ro/courses/pm/proiect/xplainedmini>
- **Schema si Cablaj PM-Board:** <https://github.com/dantudose/PM-Board>

## Resurse Software

- **MFRC522 Library:** <https://github.com/miguelbalboa/rfid>
- **SPI Library:** <https://github.com/arduino/ArduinoCore-avr/tree/master/libraries/SPI>
- **Arduino Core for AVR Framework:** <https://github.com/arduino/ArduinoCore-avr>
- **AVR Libc (util/delay.h):** [https://www.nongnu.org/avr-libc/user-manual/group\\_util\\_delay.html](https://www.nongnu.org/avr-libc/user-manual/group_util_delay.html)

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2026/ciprian.popescu0411/alexandru.dabija04>



Last update: **2026/05/24 18:33**